ソフトウェア信頼度成長モデルに基づく最適リリース問題(計画数学とその関連分野)

山田, 茂; 尾崎, 俊治

数理解析研究所講究録
第680巻1989年49-60

ソフトウェア信頼度成長モデルに基づく最適リリース問題

Optimal Release Problems Based on Software Reliability Growth Models

山田 茂 尾崎 俊治
Shigeru YAMADA Shunji OSAKI
広島大学工学部第二類（電気系）
(Faculty of Engineering, Hiroshima University)

1 まえがき

ソフトウェア開発の最終段階であるテスト工程やソフトウェアが出荷された後の運用段階において，品質・信頼性評価を定量的に行うために多くの信頼性モデルが研究されている．元来ソフトウェアは，人間の作った論理を表現した知的生産物であるため，多くの人為的誤りや欠陥，いわゆるソフトウェアエラー(以下エラーと略す)が開発過程の各段階で作り込まれ，エラーによるソフトウェアの障害や故障の発生が不確定現象となって表面化する．このため，ソフトウェア開発のテストはエラーの発見と修正を目的として行われている．このテスト工程におけるエラーの発見事象や故障発生現象を，確率・統計論に基づいて信頼性モデルとして記述するソフトウェア信頼度成長モデル (software reliability growth model)[(1),(2)] が信頼性評価のためによく用いられている．ここで，ソフトウェア故障とはソフトウェア内の潜在するエラーにより期待通りにソフトウェアが動作しないことと定義する．このモデルは，テストに費やされた時間と発見された総エラー数あるいは発生した総故障数の関係を信頼度成長過程と見なすものである．これまでに提案されてきたソフトウェア信頼度成長モデルを分類すると，時間計測モデル，個数計測モデル，およびアベイラビリティモデルの3つに大別される[(3)]．このうち本論文では，発生したソフトウェア故障数あるいは発見されたエラー数に基づく個数計測モデルと，ソフトウェア故障発生時間あるいはエラー発見時間に基づく時間計測モデルを取り扱う．

一般に，ソフトウェア信頼度成長モデルに基づいてテスト工程での観測データを解析することにより，ソフトウェア内の総期待エラー数，平均ソフトウェア故障発生時間間隔，ソフトウェア信頼度などの信頼性評価尺度が推定・予測できる．この結果により，テストの進捗状況を把握して，ユーザにソフトウェアを引き渡すのに最適なリリース (release) 時期を見積ることも可能となる．

本論文では，個数計測モデルとして代表的な非同次ポアソン過程により記述される5つのモデル[(4)]と，時間計測モデルとして代表的な Jelinski and Moranda のモデル[(5)]および Schick and Wolverton のモデル[(6)]に基づくソフトウェアの最適リリース問題について議論する．これらのソフトウェア信頼度成長モデルの概要を述べた上で，コスト評価基準による個数計測モデルおよび時間計測モデルに基づく最適リリース問題を定式化し，それぞれの最適解を与える．コスト評価基準により導出される評価関数は，テスト工程および運用段階で発生するコスト要因を考慮した総期待ソフトウェアコストであり，これを最小化する総テスト時間を最適リリース時刻とする．

2 ソフトウェア信頼度成長モデル

ソフトウェア開発のテスト工程では，所定の手順に従ってソフトウェア内に潜在するエラーを発見し，それを修正する．ここで，発見されたエラーの修正時には新しいエラーは作り込ま

れないものと仮定する．このとき，テストの進行と共にソフトウェア内に残存する総エラー数は減少していくので，ソフトウェア故障の起こらない確率，すなわちソフトウェアの信頼度は増加することになる．このようなテスト工程におけるソフトウェア故障発生現象を記述し，信頼性評価に用いられる数理モデルはソフトウェア信頼度成長モデルと呼ばれる．

ソフトウェア信頼度成長モデルは，通常確率・統計論を使って故障発生現象をモデル化するので，山田[3]は故障特性を表す確率変数に何をとるかにより個数計測モデル，時間計測モデル，およびアベイラビリティモデルの3つに大別した．このうち本論文では，発生したソフトウェア故障数や発見されたエラー数に基づく個数計測モデルと，ソフトウェア故障発生時間やエラー発見時間に基づく時間計測モデルを取り上げる．

2.1 個数計測モデル

代表的な個数計測モデルは，非同次ポアソン過程 (nonhomogeneous Poisson process，以下 NHPP と略す) を確率則として導入したソフトウェア信頼度成長モデルである (Yamada and Osaki[4]参照)．テスト時刻 t までに発見された累積エラー数を表す計数過程 $\{N(t), t \geq 0\}$ を考える．この計数過程に NHPP を仮定すると，ソフトウェア信頼度成長モデルは

$$Pr\{N(t)=n\} = \frac{\{H(t)\}^n}{n!}\exp[-H(t)] \qquad (n=0,1,2,\ldots), \tag{1}$$

と定式化される．ここで，$H(t)$ は平均値関数と呼ばれる $N(t)$ の期待値 $E[N(t)]$ であり，テスト時刻 t までに発見される総期待エラー数を意味する．テスト開始前にソフトウェア内に潜在する総期待エラー数を a とすると，テスト時刻 t においてソフトウェア内に残存するエラー数の期待値は，$a = E[N(\infty)]$ であるから

$$n_r(t) = a - H(t), \tag{2}$$

となる．さらに，テスト時刻tでソフトウェア故障が起こったという条件の下で，時間区間 $(t, t+x]$ においてソフトウェア故障の起こらない条件付き確率は，NHPP の性質から

$$R(x \mid t) = \exp[-\{H(t+x) - H(t)\}] \qquad (t \geq 0, x \geq 0), \tag{3}$$

により与えられる．式 (3) は，ソフトウェア信頼度 (software reliability) と呼ばれている．式 (2) の期待残存エラー数や式 (3) のソフトウェア信頼度は，ソフトウェア信頼性評価のための定量的尺度として使うことができる．

式 (1) の平均値関数に具体的な関数形を与えることにより，信頼性評価モデルが定式化される．ここでは，テスト工程における実際のエラー発見事象を反映した，NHPP モデルを考える．なお，費やされたテスト時間と，そのときまでに発見された累積エラー数との関係を表す時間的傾向は，その成長曲線により知ることができる．この成長曲線が指数形およびS字形を示すエラー発見事象を記述する信頼性モデルは，それぞれ指数形ソフトウェア信頼度成長モデルおよびS字形ソフトウェア信頼度成長モデルと呼ばれる．

Goel and Okumoto[7]は，単位時間当りに発見されるエラー数はその時刻において残存するエラー数に比例するものと仮定して，

$$H(t) \equiv m(t) = a(1 - \mathrm{e}^{-bt}) \qquad (a>0, b>0), \tag{4}$$

とする指数形ソフトウェア信頼度成長モデルを提案した．ここで，パラメータ b は任意のテスト時刻において残存するエラー1個当りのエラー発見率を表す．

Yamada and Osaki[8]は，テストにより発見されるエラーには発見容易なエラー (タイプ1エラー) と発見困難なエラー (タイプ2エラー) の2種類があるものと仮定して，

$$H(t) \equiv m_p(t) = a\sum_{i=1}^{2} p_i(1-\mathrm{e}^{-b_i t}) \quad (a>0, 0<b_2<b_1<1, \sum_{i=1}^{2} p_i = 1, 0<p_i<1), \tag{5}$$

とする修正指数形ソフトウェア信頼度成長モデルを提案した．ここで，パラメータb_iおよびp_iは，それぞれタイプiエラーの1個当りのエラー発見率およびエラー含有率を表す$(i=1,2)$．

山田・尾崎[9]および Yamada et al.[10]は，テストの実行過程がソフトウェア故障の現象を観測する過程(故障発見)とその原因解析を行ってエラーの発見に至る過程(エラー認知)という2つの過程から成ると考え，

$$H(t) \equiv M(t) = a[1-(1+bt)\mathrm{e}^{-bt}] \qquad (a>0, b>0), \tag{6}$$

とする遅延S字形ソフトウェア信頼度成長モデルを提案した．ここで，パラメータbは定常状態における1個当りのエラー発見率を表す．

Ohba[11]および Yamada et al.[12]は，テストチームのソフトウェアに対する習熟度を考慮し，テストによりエラーは互いに独立には発見されないものと仮定して，

$$H(t) \equiv a(1-\mathrm{e}^{-bt})/(1+c\cdot\mathrm{e}^{-bt}) \qquad (a>0, b>0, c>0), \tag{7}$$

とする習熟S字形ソフトウェア信頼度成長モデルを提案した．ここで，パラメータbは定常状態における1個当りのエラー発見率，パラメータcは習熟係数を表す．

以上4つのNHPPに基づくソフトウェア信頼度成長モデルでは，ソフトウェアの信頼性に密接に関係するテスト労力の影響を考慮していない．テスト労力とは，ソフトウェア開発のテスト工程で投入されるテスト資源であり，工数(マンパワー)，CPU時間，実行されたテストケース数などにより計測される．Yamada et al.[13]は，任意のテスト時刻におけるテスト労力の投入量に対して単位時間当りに発見されるエラー数は，そのテスト時刻での残存エラー数に比例するものと仮定し，

$$\left.\begin{array}{ll} H(t) \equiv Z(t) = a(1-\exp[-rW(t)]) & (a>0, 0<r<1) \\ W(t) = \alpha(1-\exp[-\beta t^m]) & (\alpha>0, \beta>0, m>0) \end{array}\right\}, \tag{8}$$

として，テスト工程に投入されるテスト労力を信頼性に関係づけるソフトウェア信頼度成長モデルを提案した．このモデルを，テスト労力依存型ソフトウェア信頼度成長モデルと呼ぶ．ここで，パラメータrは投入された単位テスト労力当りのエラー発見率，$W(t)$はテスト時刻tまでに投入される総テスト労力を表すワイブル(Weibull)型テスト労力関数，パラメータα, β, およびmはワイブル型テスト労力関数$W(t)$を決定する定数パラメータである．特に，パラメータαはテスト工程で必要とされる総テスト労力量を表す．

2.2 時間計測モデル

ソフトウェアのテストを開始してから，$(i-1)$番目とi番目の間のソフトウェア故障発生時間間隔を表す確率変数をX_iとし，その累積分布関数を$F(x_i)$とする$(i=1,2,\ldots,n)$．このとき，X_iに対する瞬間故障率(hazard rate)は，

$$z(x_i) \equiv \frac{\frac{d}{dx_i}F(x_i)}{1-F(x_i)}, \tag{9}$$

により定義される．代表的な2つの時間計測モデルを考える．

Jelinski and Moranda[5]は，瞬間故障率$z(x_i)$がソフトウェア内に残存するエラー数に比例し，テスト時間に関して一定であると仮定して，

$$z(x_i) = \phi[N-(i-1)] \qquad (i=1,2,\ldots,N; N>0, \phi>0), \tag{10}$$

となるソフトウェア信頼度成長モデルを提案した．ここで，パラメータ N はテスト開始前にソフトウェア内に潜在する総エラー数，パラメータ ϕ はソフトウェア故障1回当りまたはエラー1個当りの発見率を表す．式 (10) から，確率変数 X_i に対するソフトウェアの信頼度および平均故障時間間隔は，それぞれ

$$R(x_i) = \exp[-\int_0^{x_i} z(x_i)dx_i] = \exp[-\phi(N-i+1)x_i], \tag{11}$$

$$E[X_i] = \int_0^{\infty} R(x_i)dx_i = 1/[(N-i+1)\phi], \tag{12}$$

により与えられる．

Schick and Wolverton(6) は，式 (10) において瞬間故障率が実施されたテスト時間にも関係するものと仮定して，

$$z(x_i) = \phi[N-(i-1)]x_i \quad (i=1,2,\ldots,N; N>0, \phi>0), \tag{13}$$

となるソフトウェア信頼度成長モデルを提案した．このモデルでは，パラメータ ϕ は単位時間当りの (ソフトウェア故障1回当りの) 故障率を表す．式 (13) から，確率変数 X_i に対するソフトウェアの信頼度および平均故障時間間隔は，それぞれ

$$R(x_i) = \exp[-\phi(N-i+1){x_i}^2/2], \tag{14}$$

$$E[X_i] = \{\pi/[2(N-i+1)\phi]\}^{\frac{1}{2}}, \tag{15}$$

により与えられる．

3 ソフトウェアのコスト・パラメータ

ソフトウェアの信頼性データ解析を行なった後に興味ある問題の1つに，ソフトウェア開発のテストを終了して次の運用段階に移行するのに最適な時期を見積ることがある．これは，引き続きテストを継続するべきか，それともユーザにソフトウェアを引き渡すべきかを決定する問題として定式化され，ソフトウェアの最適リリース問題 (optimal software release problem) と呼ばれている(14)−(18)．最適リリース問題を定式化するにあたり，評価基準として特定の信頼性評価尺度や関係する総期待コストなどが採用されている．前者では，例えば式 (2) の期待残存エラー数や式 (3) のソフトウェア信頼度などが，目標値を達成するのに必要な総テスト時間を求めることになる．本論文では，後者のコスト評価基準を採用し，テスト工程および運用段階で発生するコスト要因を 2. で考察したソフトウェア信頼度成長モデルに基づいて分析する．そして，総期待ソフトウェアコストを求めて，これを最小化するような総テスト時間を最適リリース時刻として議論する．

まず，次のコスト・パラメータを定義する．

c_1=テスト工程において発見されるエラー1個当りの修正コスト，
c_2=運用段階において発見されるエラー1個当りの修正コスト，
c_3 = 単位テスト時間当りのコスト．

ここで，$c_2 > c_1 > 0, c_3 > 0$ である．

4 個数計測モデルに対する最適リリース問題

式 (1) で定義されるような，平均値関数 $H(t)$ をもつ NHPP に基づくソフトウェア信頼度成長モデルを考える．ここで，総テスト時間は T であり，テスト開始時点から計測したソフトウェアのライフサイクルの長さを T_{LC} とする．$H(T)$ はテスト時刻 T までに発見される総エラー数の期待値であるから，テスト工程および運用段階において発見されるエラーを修正するのに要する期待コストは，それぞれ $c_1H(T)$ および $c_2\{H(T_{LC}) - H(T)\}$ となる．さらに，T 時間のテストに要する期待コストは c_3T である．したがって，総期待ソフトウェアコストは

$$C(T, T_{LC}) = c_1H(T) + c_2\{H(T_{LC}) - H(T)\} + c_3T\ , \tag{16}$$

となる．式 (16) の $C(T, T_{LC})$ を最小にする最適解 $T = T^*$ が最適リリース時刻である．

具体的に，式 (4)–式 (7) の平均値関数のうち，式 (4) の平均値関数 $m(t)$ をもつ指数形ソフトウェア信頼度成長モデルを仮定したときの最適リリース問題を考える．式 (16) の $H(t)$ に式 (4) の $m(t)$ を代入すると

$$C(T, T_{LC}) = c_1m(T) + c_2\{m(T_{LC}) - m(T)\} + c_3T, \tag{17}$$

となる．式 (17) より，$dC(T, T_{LC})/dT = 0$ とおくと

$$h_m(T) = \frac{c_3}{c_2 - c_1}\ , \tag{18}$$

を得る．式 (18) の $h_m(\mathrm{t})$ は，平均値関数 $m(t)$ をもつ NHPP の強度関数 (intensity function)

$$h_m(T) \equiv \frac{dm(T)}{dT} = abe^{-bT},\ h_m(0) = ab, \tag{19}$$

であり，単位テスト時間当りに発見されるエラー数を意味する．式 (19) の $h_m(t)$ は，総テスト時間 T に関して単調減少関数であることに注意する．このとき，$h_m(0) \le c_3/(c_2 - c_1)$ ならば式 (18) には有限な解がなく，$dC(T, T_{LC})/dT > 0$ $(T > 0)$ となる．したがって，式 (17) の $C(T, T_{LC})$ は $T = 0$ のとき最小となり $T^* = 0$ である．すなわち，テストを実施せずにユーザにソフトウェアをリリースすることになる．また，$h_m(0) > c_3/(c_2 - c_1)$ ならば式 (18) の有限かつ唯一の解が存在し，この解は，

$$T_0 = \frac{1}{b}\ln[\frac{ab(c_2 - c_1)}{c_3}], \tag{20}$$

により与えられる．したがって，$0 < T < T_0$ のとき $dC(T, T_{LC})/dT < 0$ かつ $T > T_0$ のとき $dC(T, T_{LC})/dT > 0$ であるから，$T_0 \le T_{LC}$ のとき $T = T_0$ で $C(T, T_{LC})$ は最小となり，$T_0 > T_{LC}$ のとき $T = T_{LC}$ で $C(T, T_{LC})$ は最小となる．ゆえに，$T^* = \min\{T_0, T_{LC}\}$ となる．

以上をまとめて次の定理を得る．

［定理 1］ $c_2 > c_1 > 0, c_3 > 0$ とする．

(1) もし $h_m(0) > c_3/(c_2 - c_1)$ ならば，式 (18) の有限かつ唯一の解 $T = T_0$ が存在し，最適リリース時刻は

$$T^* = \min\{T_0, T_{LC}\},$$

である．T_0 は式 (20) により与えられる．

(2) もし $h_m(0) \leq c_3/(c_2-c_1)$ ならば，最適リリース時刻は $T^*=0$ である．

一方，式 (8) で定義されるような，テスト工程におけるテスト労力の投入量と信頼性を関係づけたソフトウェア信頼度成長モデルに基づく最適リリース問題を考える．ここで，コスト・パラメータのうち c_3を，新たに次のように定義し，コスト・パラメータ c_1および c_2は上記と同一であるとする．

c_3 =単位テスト労力投入量当りのコスト．

このとき，テスト労力の投入量に対する期待コストは $c_3W(T)$ となり，式 (8) の平均値関数 $Z(t)$ をもつテスト労力依存型ソフトウェア信頼度成長モデルに対する総期待ソフトウェアコストは

$$C(T,T_{LC}) = c_1Z(T) + c_2\{Z(T_{LC}) - Z(T)\} + c_3W(T), \tag{21}$$

となる．式 (21) より，$dC(T,T_{LC})/dT = 0$ とおくと

$$E(T) = \frac{c_3}{c_2-c_1}, \tag{22}$$

を得る．ここで，

$$E(T) \equiv r[a - Z(T)],\ E(0) = ar, \tag{23}$$

とする．式 (23) の $E(T)$ は，単位テスト労力投入量当りに発見されるエラー数を意味し，総テスト時間 Tに関して単調減少関数である．このとき，

$$T_0 = \sqrt[m]{-\frac{1}{\beta}\ln\{1-\frac{1}{r\alpha}\ln[\frac{ar(c_2-c_1)}{c_3}]\}}\ , \tag{24}$$

とすれば，定理 1 と同様にして次の定理を得る．

［定理 2］ $c_2 > c_1 > 0$, $c_3 > 0$ とする．

(1) もし $E(0) \leq c_3/(c_2-c_1)$ ならば $T^*=0$ である．

(2) もし $E(0) > c_3/(c_2-c_1) > E(T_{LC})$ ならば $T^*=T_0$であり，T_0は式 (24) により与えられる．

(3) もし $E(T_{LC}) \geq c_3/(c_2-c_1)$ ならば $T^*=T_{LC}$である．

これまではコスト評価基準を使って，NHPP に基づくソフトウェア信頼度成長モデルにより，ソフトウェアの最適リリース問題を議論してきた．実際には，関係する総期待ソフトウェアコストの最適化を図るだけでなく，実施されたテストの結果，達成される信頼度などの信頼性評価尺度をも考慮すべきである．そこで，式 (3) のソフトウェア信頼度と式 (16) の総期待ソフトウェアコストという 2 つの評価基準を同時に考慮した最適リリース問題を考える[19],[20]．このとき，テストを実施することにより向上していく式 (3) のソフトウェア信頼度 $R(x|T)$ が目標信頼度 R_0を満足しながら，式 (16) の総期待ソフトウェアコスト $C(T,T_{LC})$ を最小にするものとする．したがって，このような最適リリース時刻は，指定された運用時間 $x (x \geq 0)$ に対して，

$$\left.\begin{array}{l}\text{minimize } C(T,T_{LC})\\ \text{subject to } R(x|T)\ge R_0, T\ge 0\end{array}\right\}, \tag{25}$$

を満足する総テスト時間 $T=T^*$を求める問題として定式化される．式(4)の平均値関数 $m(t)$ を持つ指数形ソフトウェア信頼度成長モデルでは，式(25)は

$$\text{minimize } [c_1m(T)+c_2\{m(T_{LC})-m(T)\}+c_3T], \tag{26}$$

$$\text{subject to } \exp[-\{m(T+x)-m(T)\}]\ge R_0,\ T\ge 0, \tag{27}$$

となる．式(27)のソフトウェア信頼度は，総テスト時間Tに関して $R(x|0)=\exp[-m(x)]$ および $R(x|\infty)=1$ の単調増加関数であるので，$R(x|0)<R_0<1$ ならば $R(x|T)=R_0$を満足する有限かつ唯一の解が存在し，

$$T_1=\frac{1}{b}\ln[m(x)/\ln(\frac{1}{R_0})], \tag{28}$$

により与えられる．したがって，定理1の結果を使って，式(26)および式(27)を満足するソフトウェアの最適リリース時刻に関する次の定理を得る．

［定理3］ $c_2>c_1>0, c_3>0, x\ge 0, 0<R_0<1$ とする．

(1) もし $h_m(0)>c_3/(c_2-c_1)$ かつ $R(x|0)<R_0$ならば，それぞれ式(20)および式(28)を満足する有限かつ唯一の解 $T=T_0$および $T=T_1$が存在し，最適リリース時刻は

$$T^*=\max\{T_0,T_1\},$$

である．

(2) もし $h_m(0)>c_3/(c_2-c_1)$ かつ $R(x|0)\ge R_0$ならば，最適リリース時刻は $T^*=T_0$である．

(3) もし $h_m(0)\le c_3/(c_2-c_1)$ かつ $R(x|0)<R_0$ならば，最適リリース時刻は $T^*=T_1$である．

(4) もし $h_m(0)\le c_3/(c_2-c_1)$ かつ $R(x|0)\ge R_0$ならば，最適リリース時刻は $T^*=0$ である．

定理3に於て，最適リリース問題の実用性を考えて $T_{LC}>\max\{T_0,T_1\}$ と仮定した．

5　時間計測モデルに対する最適リリース問題

テストによりn個のエラーが発見された時点で，ソフトウェアをユーザにリリースするものと仮定する．したがって，2．2からソフトウェアのリリース時刻の平均値は，

$$\sum_{i=1}^{n}E[X_i],$$

となる．テスト工程においては，発見されたエラーの修正・除去に伴うコスト要因とテスト時間に比例するコスト要因を考えると，その期待コストは

$$c_1n+c_3\sum_{i=1}^{n}E[X_i],$$

である．運用段階においては，テストにより発見されなかったエラーによるソフトウェア故障の発生に対する保守コストが必要となるので，その期待コストは

$$c_2(N-n),$$

である．したがって，総期待ソフトウェアコストは

$$C(n) = c_1 n + c_2(N-n) + c_3 \sum_{i=1}^{n} E[X_i], \tag{29}$$

となる．式 (29) の $C(n)$ を最小にする $n = n^*$ が求まると，ソフトウェアの最適リリース時刻 T^* は

$$T^* = \sum_{i=1}^{n^*} E[X_i], \tag{30}$$

により与えられる．式 (29) は，Jelinski and Moranda のモデルおよび Schick and Wolverton のモデルに対して，式 (12) および式 (15) より，それぞれ

$$C(n) = c_1 n + c_2(N-n) + c_3 \sum_{i=1}^{n} 1/[(N-i+1)\phi], \tag{31}$$

$$C(n) = c_1 n + c_2(N-n) + c_3 \sum_{i=1}^{n} \{\pi/[2(N-i+1)\phi]\}^{\frac{1}{2}}, \tag{32}$$

となる．

式 (31) の $C(n)$ を最小にする $n = n^*$ を求めるために，

$$Y(n) = -(c_2 - c_1) + c_3/[(N-n)\phi], \tag{33}$$

を定義すると，$Y(n)$ は発見エラー数 n に関して

$$Y(0) = -(c_2 - c_1) + c_3/(N\phi),\ Y(N) = \infty, \tag{34}$$

の単調増加関数であることがわかる．このとき，$N\phi > c_3/(c_2 - c_1)$ ならば $Y(0) < 0$ となり，$Y(n) \geq 0$ となる最小の n に対して $C(n+1) \geq C(n)$ かつ $C(n) < C(n-1)$ $(n = 1, 2, \ldots)$ を満足する．したがって，式 (33) より $Y(n) = 0$ の有限かつ唯一の解

$$n_0 = \lceil N - c_3/\{(c_2 - c_1)\phi\} \rceil, \tag{35}$$

は，式 (31) の $C(n)$ を最小にする最適解 n^* である．ここで，$\lceil x \rceil$ は x 以上の最小の整数値を表わす．一方，$N\phi \leq c_3/(c_2 - c_1)$ ならば $Y(0) \geq 0$ となり，$C(n+1) \geq C(n) (n = 0, 1, 2, \ldots)$ となるから，式 (31) の $C(n)$ を最小にする最適解は $n^* = 0$ である．すなわち，テストを実施せずにユーザにソフトウェアをリリースすることになる．以上の Jelinski and Moranda のモデルに対する最適リリース時刻に関して，次の定理を得る．

［定理 4］ $c_2 > c_1 > 0, c_3 > 0$ とする．

(1) もし $N\phi > c_3/(c_2 - c_1)$ ならば，式 (31) の $C(n)$ を最小にする有限かつ唯一の最適解 $n^* = n_0$ が存在し，最適リリース時刻は

$$T^* = \sum_{i=1}^{n_0} E[X_i],$$

となる．n_0 は式 (35) により与えられる．

(2) もし $N\phi \leq c_3/(c_2 - c_1)$ ならば，最適リリース時刻は $T^* = 0$ である．

次に，式(32)の$C(n)$を最小にする$n=n^*$を求めるために，定理4と同様にして，

$$n_0 = \lceil N - \frac{\pi}{2\phi}(\frac{c_3}{c_2-c_1})^2 \rceil, \tag{36}$$

とすればSchick and Wolvertonのモデルに対する最適リリース時刻に関して，次の定理を得る．

［定理5］ $c_2 > c_1 > 0, c_3 > 0$とする．

(1) もし$N\phi > \pi\{c_3/(c_2-c_1)\}^2/2$ならば，式(32)の$C(n)$を最小にする有限かつ唯一の最適解$n^* = n_0$が存在し，最適リリース時刻は

$$T^* = \sum_{i=1}^{n_0} E[X_i],$$

となる．n_0は式(36)により与えられる．

(2) もし$N\phi \le \pi\{c_3/(c_2-c_1)\}^2/2$ならば，最適リリース時刻は$T^*=0$である．

時間計測モデルの場合と同様にして，コストおよび信頼性評価基準を同時に満足するようなソフトウエアの最適リリース問題を考える．このとき，式(11)および式(14)から，所定の動作時間をxとするときにソフトウエアの信頼度の目標値を$R_0(0 < R_0 < 1)$とすれば，Jelinski and Morandaのモデルに対しては，

$$\left.\begin{array}{l} \text{minimize } [c_1 n + c_2(N-n) + c_3 \sum_{i=1}^{n} 1/[(N-i+1)\phi] \\ \text{subject to } \exp[-(N-n)\phi x] \ge R_0, n \ge 0 \end{array}\right\}, \tag{37}$$

となり，Schick and Wolvertonのモデルに対しては，

$$\left.\begin{array}{l} \text{minimize } [c_1 n + c_2(N-n) + c_3 \sum_{i=1}^{n} \{\pi/[2(N-i+1)\phi]\}^{\frac{1}{2}} \\ \text{subject to } \exp[-(N-n)\phi x^2/2] \ge R_0, n \ge 0 \end{array}\right\}, \tag{38}$$

となる発見エラー数nに関する条件付き最小化問題を得る．式(37)および式(38)において，ソフトウエアの信頼度をそれぞれ

$$R(x|n) = \exp[-(N-n)\phi x], \tag{39}$$

$$R(x|n) = \exp[-(N-n)\phi x^2/2], \tag{40}$$

とする．このとき，式(39)および式(40)の両式の$R(x|n)$は，発見エラー数nに関して単調増加関数であることがわかる．したがって，$R(x|0) < R_0$ならば$R(x|n) = R_0$を満足する有限かつ唯一の解$n = n_1$は，式(39)および式(40)に対してそれぞれ

$$n_1 = \lceil N + \ln R_0/(\phi x) \rceil, \tag{41}$$

$$n_1 = \lceil N + 2\ln R_0/(\phi x^2) \rceil, \tag{42}$$

となる．

したがって，ソフトウエアの最適リリース時刻に関して定理3と同様にすれば，Jelinski and Moranda のモデルに対しては定理6，Schick and Wolverton のモデルに対しては定理7を得る．

[定理6] $c_2 > c_1 > 0, c_3 > 0$，$x \geq 0$，$0 < R_0 < 1$ とする．

(1) もし $N\phi > c_3/(c_2 - c_1)$ かつ $\exp[-N\phi x] < R_0$ ならば，それぞれ式(35)および式(41)を満足する有限かつ唯一の解 $n = n_0$ および $n = n_1$ が存在して，式(37)の最適解は，

$$n^* = \max\{n_0, n_1\},$$

となる．したがって．最適リリース時刻は

$$T^* = \sum_{i=1}^{\max\{n_0,n_1\}} E[X_i],$$

である．

(2) もし $N\phi > c_3/(c_2 - c_1)$ かつ $\exp[-N\phi x] \geq R_0$ ならば，式(37)の最適解は $n^* = n_0$ であり，最適リリース時刻は

$$T^* = \sum_{i=1}^{n_0} E[X_i],$$

となる．

(3) もし $N\phi \leq c_3/(c_2 - c_1)$ かつ $\exp[-N\phi x] < R_0$ ならば，式(37)の最適解は $n^* = n_1$ であり，最適リリース時刻は

$$T^* = \sum_{i=1}^{n_1} E[X_i],$$

となる．

(4) もし $N\phi \leq c_3/(c_2 - c_1)$ かつ $\exp[-N\phi x] \geq R_0$ ならば，最適リリース時刻は $T^* = 0$ となる．

[定理7] $c_2 > c_1 > 0$，$c_3 > 0$，$x \geq 0$，$0 < R_0 < 1$ とする．

(1) もし $N\phi > \pi\{c_3/(c_2 - c_1)\}^2/2$ かつ $\exp[-N\phi x^2/2] < R_0$ ならば，それぞれ式(36)および式(42)を満足する有限かつ唯一の解 $n = n_0$ および $n = n_1$ が存在して，式(38)の最適解は

$$n^* = \max\{n_0, n_1\},$$

となる．したがって，最適リリース時刻は

$$T^* = \sum_{i=1}^{\max\{n_0,n_1\}} E[X_i],$$

である．

(2) もし $N\phi > \pi\{c_3/(c_2 - c_1)\}^2/2$ かつ $\exp[-N\phi x^2/2] \geq R_0$ ならば，式(38)の最適解は $n^* = n_0$ であり，最適リリース時刻は

$$T^* = \sum_{i=1}^{n_0} E[X_i],$$

となる．

(3) もし $N\phi \leq \pi\{c_3/(c_2-c_1)\}^2/2$ かつ $\exp[-N\phi x^2/2] < R_0$ ならば，式 (38) の最適解は $n^* = n_1$ であり，最適リリース時刻は

$$T^* = \sum_{i=1}^{n_1} E[X_i],$$

となる．

(4) もし $N\phi \leq \pi\{c_3/(c_2-c_1)\}^2/2$ かつ $\exp[-N\phi x^2/2] \geq R_0$ ならば，最適リリース時刻は $T^* = 0$ である．

6 むすび

本論文では，ソフトウェア信頼度成長モデルに基づいて，ソフトウェア開発のテストを終了してユーザの運用段階へ移行するのに最適な時期を見積る方法，すなわちソフトウェアの最適リリース問題について議論した．ソフトウェア信頼度成長モデルとしては，個数計測モデルとして非同次ポアソン過程により記述される5つのモデルと，時間計測モデルとして Jelinski and Moranda のモデルおよび Schick and Wolverton のモデルを取り扱った．また，最適リリース時刻を導出するための評価基準として，テスト工程および運用段階で発生するコスト要因を分析して算出した総期待ソフトウェアコストを採用し，これを最小化する総テスト時間を最適リリース時刻とした．なお，テストにより達成されるソフトウェアの信頼度も無視できないとして，コストおよび信頼度評価基準を同時に考慮した最適リリース問題も議論した．

実際問題としては，3．で導入したコスト・パラメータの見積り方法，達成されるソフトウェアの信頼度の設定の仕方，更にはユーザに対するソフトウェアの納期の考慮などが重要となる．

謝辞 有益なご助言を頂いた大阪工業大学　一森哲男助教授に感謝の意を表します．

参考文献

(1) C.V. Ramamoorthy and F.B. Bastani: "Software reliability-Status and perspectives", IEEE Trans. Software Eng.,Vol.SE-8, No.4, pp.354–371 (July 1987).

(2) J.D. Musa, A. Iannino and K. Okumoto: "Software Reliability: Measurement, Prediction, Application", McGraw-Hill, New York (1987).

(3) 山田　茂: "ソフトウェアの信頼性評価法"，ソフト・リサーチ・センター (1985).

(4) S. Yamada and S. Osaki: "Software reliability growth modeling: Models and applications", IEEE Trans. Software Eng., Vol.SE-11, No.12, pp.1431–1437 (Dec. 1985).

(5) Z. Jelinski and P.B. Moranda: "Software reliability research", in Statistical Computer Performance Evaluation, W. Freiberger (ed.), pp.465–484,Academic Press, New York (1987).

(6) G.J. Schick and R.W. Wolverton: "Analysis of competing software reliability models",IEEE Trans. Software Eng., Vol.SE-4, No.2, pp.104–120 (Mar. 1987).

(7) A.L. Goel and K.Okumoto: "Time-dependent error-detection rate model for software reliability and other performance measures", IEEE Trans. Reliability, Vol.R-28, No.3, pp.206–211 (Aug. 1979).

(8) S. Yamada and S. Osaki: "Nonhomogeneous error detection rate models for software reliability growth", in Stochastic Models in Reliability Theory, S. Osaki and Y. Hatoyama (eds.), pp.120–143, Springer-Verlag, Berlin (1984).

(9) 山田・尾崎: "ソフトウェアの信頼度成長モデルとその比較", 電子通信学会論文誌, Vol.J65-D, No.7, pp.906–912 (1982 年 7 月).

(10) S. Yamada, M. Ohba and S. Osaki: "S-shaped reliability growth modeling for software error detection",IEEE Trans. Reliability, Vol.R-32, No.5, pp.475–478,484 (Dec. 1983).

(11) M. Ohba: "Software reliability analysis models", IBM J. Research and Development, Vol.28, No.4, pp. 428–443 (July 1984).

(12) S. Yamada, M. Ohba and S. Osaki: "S-shaped software reliability growth models and their applications", IEEE Trans. Reliability, Vol.R-33, No.4, pp.289–292 (Oct. 1984).

(13) S. Yamada, H. Ohtera and H. Narihisa: "A testing-effort dependent reliability model for computer programs",Trans. IECE Japan, Vol.E69, No.11, pp.1217–1224 (Nov. 1986).

(14) E.H. Foreman and N.D. Singpurwalla: "Optimal time intervals for testing hypotheses on computer software errors",IEEE Trans. Reliability, Vol.R-28,No.3, pp. 250–253 (Aug. 1979).

(15) K. Okumoto and A.L. Goel: "Optimum release time for software systems based on reliability and cost criteria", J. Systems and Software, Vol.1, pp.315–318 (1980).

(16) J.G. Shanthikumar and S. Tufekci: "Application of a software reliability model to decide software release time", Microelectronics and Reliability, Vol.23, No1, pp. 41–59 (1983).

(17) H.S. Koch and P. Kubat: "Optimal release time for computer software",IEEE Trans. Software. Eng., Vol.SE-9, No.3, pp.323–327 (Mar. 1983).

(18) S. Yamada, H. Narihisa and S. Osaki: "Optimum release policies for a software system with a scheduled software delivery time", Int. J. Systems Science, Vol.15, No.8, pp.905–914 (Aug. 1984).

(19) S. Yamada and S. Osaki: "Cost-reliability optimal release policies for software systems", IEEE Trans. Reliability, Vol.R-34, No.5, pp.422–424 (Dec. 1985).

(20) S. Yamada and S. Osaki: "Optimal software release policies with simultaneous cost and reliability requirements", European J. Operational Research, Vol.31, No.1, pp.46–51 (July 1987).